A polynomial procedure for trimming visibly pushdown automata

Mathieu Caralp, Pierre-Alain Reynier, Jean-Marc Talbot

A polynomial procedure for trimming visibly pushdown automata

Mathieu Caralp, Pierre-Alain Reynier, and Jean-Marc Talbot

Laboratoire d'Informatique Fondamentale de Marseille
Aix-Marseille Université & CNRS, France

Abstract. We describe a polynomial procedure which, given a visibly pushdown automaton that accepts only well-nested words, returns an equivalent visibly pushdown automaton that is trimmed. We also show that this procedure can be applied to weighted visibly pushdown automata such as visibly pushdown transducers.

1 Introduction

Visibly pushdown automata (VPA) are a particular class of pushdown automata working over an alphabet splitted into call, internal and return symbols [1,2]. In VPA's, the stack behaviour is imposed by the input word: on a call symbol, the VPA pushes a symbol onto the stack, on a return symbol, it must pop the top symbol of the stack, and on an internal symbol, the stack remains unchanged.

Trimming a finite state automaton amounts to remove useless states, that is states that do not occur in some accepting computation of the automaton. This can be done easily in linear time simply by solving two reachability problems in the graph representing the automaton. However, the problem is more difficult for VPA's as the current state of a computation (called a configuration) is given by both a "control" state and a stack content.

As VPA are a restriction of pushdown automata, one can try to apply trimming procedures for pushdown automata to VPA. We are aware of two such procedures. A first one, described in [4], has an exponential time complexity. A second one relies on the translation of pushdown automata into context-free grammars [1]. More precisely, one can first translate the automaton into an equivalent context-free grammar, then eliminate from this grammar variables generating the empty language or not reachable from the start symbol, and third convert the resulting grammar into the pushdown automaton realizing its top-down analysis. This construction has a polynomial time complexity. However, in this form, it does not apply to VPA because the resulting pushdown automaton may not satisfy the condition of visibility. It is not clear whether it can be adapted to the setting of VPA.

In this work, we present a polynomial time procedure for trimming VPA that accept only well-nested words. Note that this construction could be extended to the complete class of VPA. In addition, we show how this procedure can be applied to weighted VPA, that is VPA in which transitions are also labelled by weights (such as visibly pushdown transducers for instance).

[1] We thank Géraud Sénizergues for pointing us this construction.

2 Definitions

Words and nested words Let Σ be a finite alphabet partitioned into three disjoint sets Σ_c, Σ_r and Σ_ι, denoting respectively the *call*, *return* and *internal* alphabets. We denote by Σ^* the set of (finite) words over Σ and by ϵ the empty word. The length of a word u is denoted by $|u|$. The set of *well-nested* words Σ^*_{wn} is the smallest subset of Σ^* such that $\Sigma^*_\iota \subseteq \Sigma^*_{\mathsf{wn}}$ and for all $c \in \Sigma_c$, all $r \in \Sigma_r$, all $u, v \in \Sigma^*_{\mathsf{wn}}$, $cur \in \Sigma^*_{\mathsf{wn}}$ and $uv \in \Sigma^*_{\mathsf{wn}}$. We define the height $h(u)$ of some well-nested word u by induction as follows: $h(u) = 0$ if $u \in \Sigma^*_\iota$, $h(uv) = \max(h(u), h(v))$ for $u, v \in \Sigma^*_{\mathsf{wn}}$, and $h(cur) = 1 + h(u)$. Given a family of words $w_1, w_2, \ldots, w_n$, we denote by $\Pi_{i=1}^n w_i$ the concatenation $w_1 w_2 \ldots w_n$.

*Visibly pushdown automata (*VPA*)* Visibly pushdown automata are a restriction of pushdown automata in which the stack behaviour is imposed by the input word. On a call symbol, the VPA pushes a symbol onto the stack, on a return symbol, it must pop the top symbol of the stack, and on an internal symbol, the stack remains unchanged. Formally:

Definition 1 (Visibly pushdown automata). A *visibly pushdown automaton* (VPA) on finite words over Σ is a tuple $A = (Q, I, F, \Gamma, \delta)$ where Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ the set of final states, Γ is a finite stack alphabet, $\delta = \delta_c \uplus \delta_r \uplus \delta_\iota$ the (finite) transition relation, with $\delta_c \subseteq Q \times \Sigma_c \times \Gamma \times Q$, $\delta_r \subseteq Q \times \Sigma_r \times \Gamma \times Q$, and $\delta_\iota \subseteq Q \times \Sigma_\iota \times Q$.

A *configuration* of a VPA is a pair $(q, \sigma) \in Q \times \Gamma^*$. A *run* of A on a word $u = a_1 \ldots a_l \in \Sigma^*$ from a configuration (q, σ) to a configuration (q', σ') is a finite sequence of configurations $\rho = \{(q_k, \sigma_k)\}_{0 \leq k \leq l}$ such that $q_0 = q$, $\sigma_0 = \sigma$, $q_l = q'$, $\sigma_l = \sigma'$ and for each $1 \leq k \leq l$, there exists $\gamma_k \in \Gamma$ such that either $(q_{k-1}, a_k, \gamma_k, q_k) \in \delta_c$ and $\sigma_k = \sigma_{k-1}\gamma_k$ or $(q_{k-1}, a_k, \gamma_k, q_k) \in \delta_r$ and $\sigma_{k-1} = \sigma_k \gamma_k$, or $(q_{k-1}, a_k, q_k) \in \delta_\iota$ and $\sigma_k = \sigma_{k-1}$. Depending on the case, let $t_k \in \delta$ be the corresponding transition. Then we say that the run ρ is over the sequence of transitions $(t_k)_{1 \leq k \leq l}$. We write $(q, \sigma) \xrightarrow{u} (q', \sigma')$ when there exists a run on u from (q, σ) to (q', σ'). We may omit the superscript u when irrelevant. We denote by $\perp$ the empty word on Γ.

Initial (resp. final) configurations are configurations of the form $(q, \perp)$, with $q \in I$ (resp. $q \in F$). A configuration (q, σ) is *reachable* (resp. is *co-reachable*) if there exist $u \in \Sigma^*$ and a configuration c such that c is initial and $c \xrightarrow{u} (q, \sigma)$ (resp. such that c is final and $(q, \sigma) \xrightarrow{u} c$).

Definition 2. *An automaton A is* trimmed *if it fulfills the two following conditions:*

- *every configuration of A is reachable iff it is co-reachable,*
- *every state of A belongs to a reachable configuration.*

We say that a run is accepting if it starts in an initial configuration and ends in a final configuration. A word is accepted by A iff there exists an accepting run of A on this word. The language of A, denoted by $L(A)$, it the set of words accepted by A.

Note that the model we consider differs from the one introduced in [1], for two reasons. First, there are return transitions that can be performed on the empty stack and second, the definition of final configurations does not require an empty stack. In particular, unlike in our definition, accepted words are not necessarily well-nested. However, the results we present in this work could be extended to the general class of VPA.

3 Trimming VPA

Let $A = (Q, I, F, \Gamma, \delta)$ be a VPA on the structured alphabet Σ. In this section, we define a new VPA $A' = (Q', I', F', \Gamma', \delta')$ on Σ, denoted $\mathsf{trim}(A)$, which recognizes the same language, and in addition is trimmed.

First, we define the following set:

$$\mathsf{WN} = \{(p, p', p'') \in Q^3 \mid \exists (p, \perp) \to^* (p', \perp) \to^* (p'', \perp)\}$$

This set can be computed in polynomial time. More precisely, the following set can be computed first: $\mathsf{WN}_2 = \{(p, p') \in Q^2 \mid \exists (p, \perp) \to^* (p', \perp)\}$.

WN_2 can be defined as the least set such that

- $\{(p, p) \mid p \in Q\} \subseteq \mathsf{WN}_2$,
- if $(p, p') \in \mathsf{WN}_2$ and $(p', p'') \in \mathsf{WN}_2$, then $(p, p'') \in \mathsf{WN}_2$
- if $(p, q) \in \mathsf{WN}_2$, and $\exists (q, i, q') \in \delta_\iota$, then $(p, q') \in \mathsf{WN}_2$
- if $(p, q) \in \mathsf{WN}_2$ and $\exists (p', c, \gamma, p) \in \delta_c, (q, r, \gamma, q') \in \delta_r$, then $(p', q') \in \mathsf{WN}_2$

Then, WN is obtained from WN_2 by the following property:

$$(p, p', p'') \in \mathsf{WN} \iff (p, p') \in \mathsf{WN}_2 \wedge (p', p'') \in \mathsf{WN}_2$$

We now define the four first components of the trimmed VPA A' as follows:

- $Q' = \mathsf{WN}$
- $I' = \mathsf{WN} \cap \{(i, i, f) \mid i \in I \wedge f \in F\}$
- $F' = \mathsf{WN} \cap \{(i, f, f) \mid i \in I \wedge f \in F\}$
- $\Gamma' = \Gamma \times Q \times Q$

Intuitively, the VPA A' simulates the VPA A as follows: if a run of A' goes through a state (p, q, r) with a stack σ' of height n, then the run of A' at this position mimics a run of A whose current configuration is (q, σ), with σ of height n, and such that the top symbol of σ has been pushed when reaching the state p, and will be popped when leaving the state r. Moreover, from p to r in the run of A, the height of the stack is always larger or equal to n.

It remains to define the last component δ'. We define it by its restrictions on call, return and internal symbols respectively (namely δ'_c, δ'_r and δ'_ι).

Call symbols. Let $(p_1, p_2, p_4) \in \mathsf{WN}, c \in \Sigma_c$. Then $((p_1, p_2, p_4), c, (\gamma, p_1, p_4), (q_1, q_1, q_2)) \in \delta'_c$ iff the following three conditions hold:

- $(p_2, c, \gamma, q_1) \in \delta_c$,

- $(q_1, q_1, q_2) \in \mathsf{WN}$,
- there exists a state p_3 and $r \in \Sigma_r$ such that $(q_2, r, \gamma, p_3) \in \delta_r$, and $(p_3, p_3, p_4) \in \mathsf{WN}$.

Return symbols. Let $(q_1, q_2, q_2) \in \mathsf{WN}, r \in \Sigma_r$. Then $((q_1, q_2, q_2), r, (\gamma, p_1, p_4), (p_1, p_3, p_4)) \in \delta'_r$ iff the following three conditions hold:

- $(q_2, r, \gamma, p_3) \in \delta_r$,
- $(p_1, p_3, p_4) \in \mathsf{WN}$,
- there exists a state p_2 and $c \in \Sigma_c$ such that $(p_2, c, \gamma, q_1) \in \delta_c$, and $(p_1, p_2, p_2) \in \mathsf{WN}$.

The situation of call/return transition is depicted on Figure 1, where $w_1, w_2, w_3 \in \Sigma^*_{\mathsf{wn}}$, $c \in \Sigma_c$, and $r \in \Sigma_r$.

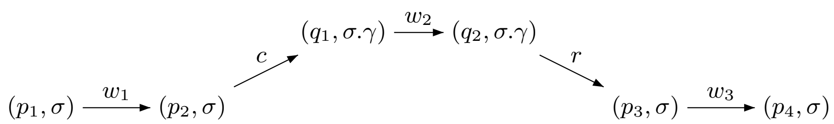

Fig. 1. Representation of call/return transitions.

Internal symbols. Let $(p_1, q, p_2), (p_1, q', p_2) \in \mathsf{WN}, a \in \Sigma_\iota$. Then $((p_1, q, p_2), a, (p_1, q', p_2)) \in \delta'_\iota$ iff the transition (q, a, q') belongs to δ_ι.

Proposition 3. *For any* VPA A*,* $\mathsf{trim}(A)$ *can be computed in polynomial time.*

This result easily follows from the definition of $\mathsf{trim}(A)$.

Proposition 4. *For any* VPA A*,* $L(A) = L(\mathsf{trim}(A))$*. More precisely, there exists a mapping* φ *from transitions of* $\mathsf{trim}(A)$ *to transitions of* A *which induces a bijection between the accepting runs of* $\mathsf{trim}(A)$ *and those of* A*.*

The mapping φ we consider is the trivial mapping obtained by considering, given a transition of A', the transition of A on which it relies. To prove this result, we will first prove different lemmas.

Lemma 5. *Let* A *be a* VPA *and* $A' = \mathsf{trim}(A)$*. Let* ρ *be a run of* A' *such that* $\rho : ((p, q, r), \sigma) \xrightarrow{w}^* ((p', q', r'), \sigma)$*, with* $\sigma \in \Gamma'^*$ *and* $w \in \Sigma^*_{\mathsf{wn}}$*. Then we have* $p = p'$ *and* $r = r'$*.*

Proof. The proof goes by induction on the structure of the word w. It holds trivially when w is the empty word. Consider now a non-empty word w. There are two cases:

- First case: $w = aw'$, with $a \in \Sigma_\iota$, $w' \in \Sigma^*_{\mathsf{wn}}$ and $((p_1, q_1, r_1), a, (p, q, r)) \in \delta'_\iota$. By construction of A', $p_1 = p$ and $r_1 = r$. We conclude by using the induction hypothesis on w'.

– Second case: $w = cw'rw''$, with $c \in \Sigma_c$, $r \in \Sigma_r$, $w', w'' \in \Sigma_{\mathsf{wn}}^*$. There exists a transition $((p_1, p_2, p_4), c, (\gamma, p_1, p_4), (q_1, q_1, q_2)) \in \delta'_c$ which is used when reading the first letter c. There exists a state (r_1, r_2, r_3) such that the run of A' on w' goes from state (q_1, q_1, q_2) to state (r_1, r_2, r_3). As w' is well-nested, the induction hypothesis applied on w' entails that $r_1 = q_1$ and $r_3 = q_2$. In addition, as a return transition is used after reading w', the definition of δ'_r implies that $r_2 = r_3$. Thus by definition of δ'_c and δ'_r, a transition of the form $((q_1, q_2, q_2), r, (\gamma, p_1, p_4), (p_1, p_3, p_4)) \in \delta'_r$ for some state p_3 is used when reading the letter r. We conclude by induction hypothesis applied on w''. □

We build a bijection between accepting runs of A and A'. First, we go from runs of A' to runs of A. We define the following standard projection mappings. Consider two integers $1 \leq j \leq k$. Given a tuple $s = (s_1, s_2, \ldots, s_k)$, we denote by $\pi_j(s)$ the element s_j. In addition, we extend this mapping over words on tuples, by letting $\pi_j(\sigma) = \pi_j(\gamma_1)\pi_j(\gamma_2)\ldots\pi_j(\gamma_n)$ where $\sigma = \gamma_1\gamma_2\ldots\gamma_n$.

Lemma 6. *Let A be a* VPA *and $A' = \mathsf{trim}(A)$. For any accepting run $\rho' = (q'_i, \sigma'_i)_{1\leq i\leq k}$ of A' over transitions $(t'_i)_{1\leq i<k}$, $\rho = (\pi_2(q'_i), \pi_1(\sigma'_i))_{1\leq i\leq k}$ is an accepting run of A over transitions $(\varphi(t'_i))_{1\leq i<k}$. In addition, if two accepting runs ρ' and ρ'' over sequences of transitions η' and η'' are projected on the same run of A (*i.e. *such that $\varphi(\eta') = \varphi(\eta'')$), then we have $\rho' = \rho''$.*

Proof. We prove by induction on ρ' that the projection run ρ defined above is a correct run of A, on the corresponding transitions. The fact that accepting runs of A' are projected on accepting runs of A is trivial. Finally, the fact that two accepting runs of $\mathsf{trim}(A)$ projected on the same run of A coincide can be proven by induction on the height of these runs. □

Conversely, considering translation of runs of A to runs of A'. States of A' extend states of A by considering starting and ending states for the current stack level, and the transitions naturally match. Therefore we prove the following lemma in which we consider a word $w_2 \in \Sigma_{\mathsf{wn}}^*$ embedded into a context $(w_1, w_3) \in \Sigma_{\mathsf{wn}}^* \times \Sigma_{\mathsf{wn}}^*$ such that $w_1w_2w_3$ corresponds to the current stack level.

Lemma 7. *Let A be a* VPA *and $A' = \mathsf{trim}(A)$. For all $w_1, w_2, w_3 \in \Sigma_{\mathsf{wn}}^*$, if there exists a run ρ of the form $(p, \bot) \xrightarrow{w_1}^* (q, \bot) \xrightarrow{w_2}^* (r, \bot) \xrightarrow{w_3}^* (s, \bot)$ in the* VPA *A, such that the middle run on w_2 is over the sequence of transitions $\eta = (t_i)_{1\leq i\leq k}$, then there is a run ρ' of the form $((p, q, s), \bot) \xrightarrow{w_2}^* ((p, r, s), \bot)$ of A' over transitions $\eta' = (t'_i)_{1\leq i\leq k}$, with $\varphi(t'_i) = t_i$ for any i.*

Proof. We prove this lemma by induction on the height of the word w_2. If $h(w_2) = 0$, then $w_2 \in \Sigma_\iota^*$, assuming $w_2 = a_1a_2\ldots a_n$, the restriction of ρ on w_2 is of the form

$$(q, \bot) \xrightarrow{a_1} (q_1, \bot) \ldots \xrightarrow{a_n} (q_n, \bot) \text{ with } r = q_n.$$

Observe that for all i, $(p, q_i, s) \in \mathsf{WN}$. By definition of A' the following is a run of A' on w_2:

$$\rho' = ((p, q, s), \bot) \xrightarrow{a_1} ((p, q_1, s), \bot) \ldots \xrightarrow{a_n} ((p, q_n, s), \bot) \text{ with } r = q_n.$$

We now assume for the induction that the property holds when $h(w_2) \leq n$ and consider w_2 such that $h(w_2) = n+1$. There exists a unique decomposition of w_2 as follows:

$$w_2 = [\Pi_{i=1}^{k}(w_i^{\mathsf{int}} c_i w_i^{\mathsf{wn}} r_i)] w_{k+1}^{\mathsf{int}}$$

with for all i, $w_i^{\mathsf{int}} \in \Sigma_\iota^*$, $w_i^{\mathsf{wn}} \in \Sigma_{wn}^*$ such that $h(w_i^{\mathsf{wn}}) \leq n$, $c_i \in \Sigma_c$, and $r_i \in \Sigma_r$.

Let $w_1, w_3 \in \Sigma_{wn}^*$ and ρ be a run of A on $w_1 w_2 w_3$. We decompose the run ρ as follows:

- on w_1: $(p, \perp) \xrightarrow{w_1}^* (p_1^1, \perp)$
- on each $w_i^{\mathsf{int}} c_i w_i^{\mathsf{wn}} r_i$:

$$(p_i^1, \perp) \xrightarrow{w_i^{\mathsf{int}}}^* (p_i^2, \perp) \xrightarrow{c_i} (q_i^1, \gamma_i) \xrightarrow{w_i^{\mathsf{wn}}}^* (q_i^2, \gamma_i) \xrightarrow{r_i} (p_{i+1}^1, \perp)$$

- on w_{k+1}^{int}: $(p_{k+1}^1, \perp) \xrightarrow{w_{k+1}^{\mathsf{int}}}^* (p_{k+1}^2, \perp)$
- on w_3: $(p_{k+1}^2, \perp) \xrightarrow{w_3}^* (s, \perp)$

Note that $q = p_1^1$ and $r = p_{k+1}^2$.

As w_i^{wn} is well-nested, there exists a run in A of the form $(q_i^1, \perp) \xrightarrow{w_i^{\mathsf{wn}}}^* (q_i^2, \perp)$. By induction hypothesis, this implies that there exists a run $\rho_i' = ((q_i^1, q_i^1, q_i^2), \perp) \xrightarrow{w_i^{\mathsf{wn}}}^* ((q_i^1, q_i^2, q_i^2), \perp)$ of A' on w_i^{wn}. Again, using the fact that w_i^{wn} is well-nested, we have $((q_i^1, q_i^1, q_i^2), \gamma_i') \xrightarrow{w_i^{\mathsf{wn}}}^* ((q_i^1, q_i^2, q_i^2), \gamma_i')$ in A' for any $\gamma_i' \in \Gamma'$.

We now describe the other parts of the run ρ' of A' on w_2.

Internal actions For all $i \in \{1, \ldots, k+1\}$, $((p, p_i^1, s), \perp) \xrightarrow{w_i^{\mathsf{int}}}^* ((p, p_i^2, s), \perp)$ on w_i^{int}, by the base induction with w_1 and w_3 well defined.

Calls For all $i \in \{1, \ldots, k\}$, we have $(p_i^2, c_i, \gamma_i, q_i^1) \in \delta_c$ in A. Thus by definition of A', there exists a transition $((p, p_i^2, s), c_i, (\gamma_i, p, s), (q_i^1, q_i^1, q_i^2)) \in \delta_c'$ in A'.

Returns For all $i \in \{1, \ldots, k\}$, we have $(q_i^2, r_i, \gamma_i, p_{i+1}^1) \in \delta_r$ in A. Thus by definition of A' there exists a transition $((q_i^1, q_i^2, q_i^2), r_i, (\gamma_i, p, s), (p, p_{i+1}^1, s)) \in \delta_r'$ in A'.

One can then easily check that all the above transitions and partial runs can be gathered to obtain a run of A' on w_2 with the expected form. $\square$

To conclude the proof of Proposition 4, note that Lemma 6 implies that $L(\mathsf{trim}(A)) \subseteq L(A)$ (more precisely, φ induces an injection from accepting runs of $\mathsf{trim}(A)$ to accepting runs of A). Conversely, using Lemma 7 for $w_1 = w_3 = \epsilon$ yields $p = q$ ans $r = s$ and if moreover p is initial and q is final then any accepting run in A is associated with an accepting run in $\mathsf{trim}(A)$. More precisely, we obtain an injection from accepting runs of A to those of $\mathsf{trim}(A)$. Indeed, thanks to the property of mapping φ, two different accepting runs of A are mapped onto two different accepting runs of $\mathsf{trim}(A)$. Finally, we obtain that mapping φ induces a bijection between runs of A and those of $\mathsf{trim}(A)$, as expected. In particular, we have $L(A) = L(\mathsf{trim}(A))$.

Proposition 8. *For any* VPA A, $\mathsf{trim}(A)$ *is trimmed.*

Proof. Let A' be $\mathsf{trim}(A)$. Therefore, we first prove that any reachable configuration of A' is also co-reachable, and then the converse.

Let $co = ((p, q, r), \sigma)$ be a reachable configuration of A'. There exists a run ρ of A' of the form $((i, i, f), \bot) \to^* co$ between an initial configuration $((i, i, f), \bot)$ and co. We show by induction on the height of the stack that we can reach a final configuration from co.

If $|\sigma| = 0$, by the lemma 5, we obtain $p = i$ and $r = f$. In particular, this implies $p \in I$ and $r \in F$. Since $(p, q, r) \in \mathsf{WN}$, there is a run $(p, \bot) \to^* (q, \bot) \to^* (r, \bot)$ in A, thus by Lemma 7 we have a run $((p, q, r), \bot) \to^* ((p, r, r), \bot)$ of A'. This concludes this case as by the above observation, we have $(p, r, r) \in I \times F \times F = F'$.

We now assume for the induction that the property holds when $|\sigma| \leq n$ and we consider a stack σ such that $|\sigma| = n + 1$. Let denote by (γ, p', r') the top symbol of σ, write $\sigma = \sigma'.(\gamma, p', r')$, and consider the first position in the run ρ that pushes this symbol onto the stack. We denote by c the associated call. More precisely, there exists a unique decomposition of ρ as follows:

$$((i, i, f), \bot) \to^* ((p', q', r'), \sigma') \xrightarrow{c} ((p'', p'', r''), \sigma) \to^* ((p, q, r), \sigma)$$

such that the run from $((p'', p'', r''), \sigma)$ to $((p, q, r), \sigma)$ is associated with a well-nested word. By Lemma 5, we obtain $p'' = p$ and $r'' = r$. Considering the call transition associated with c, and by definition of δ'_c and δ'_r, there exists a return transition $((p, r, r), r, (\gamma, p', r'), (p', q'', r')) \in \delta'_r$ for some letter $r \in \Sigma_r$. In addition, as $(p, q, r) \in \mathsf{WN}$, there is a run $(p, \sigma) \to^* (q, \sigma) \to^* (r, \sigma)$ in A, and thus by Lemma 7 we have a run $((p, q, r), \sigma) \to^* ((p, r, r), \sigma)$. As the top symbol of σ is (γ, p', r'), the above return transition can be used to reach configuration $((p', q'', r'), \sigma')$ whose height is n. The result follows by induction hypothesis.

Conversely, let $co = ((p, q, r), \sigma)$ be a co-reachable configuration of A'. There exists a run ρ of A' of the form $co \to^* ((i, f, f), \bot)$ between co and a final configuration $((i, f, f), \bot)$. We show by induction on the height of the stack that co can be reached from an initial configuration.

If $|\sigma| = 0$, by the lemma 5, we obtain $p = i$ and $r = f$. In particular, this implies $p \in I$ and $r \in F$. Since $(p, q, r) \in \mathsf{WN}$, there is a run $(p, \bot) \to^* (q, \bot) \to^* (r, \bot)$ in A, thus by Lemma 7 we have a run $((p, p, r), \bot) \to^* ((p, q, r), \bot)$ of A'. This concludes this case as by the above observation, we have $(p, p, r) \in I \times I \times F = I'$.

We now assume for the induction that the property holds when $|\sigma| \leq n$ and we consider a stack σ such that $|\sigma| = n + 1$. Let denote by (γ, p', r') the top symbol of σ, write $\sigma = \sigma'.(\gamma, p', r')$, and consider the first position in the run ρ that pops this symbol from the stack. We denote by r the associated return. More precisely, there exists a unique decomposition of ρ as follows:

$$((p, q, r), \sigma) \to^* ((p'', r'', r''), \sigma) \xrightarrow{r} ((p', q', r'), \sigma') \to^* ((i, f, f), \bot)$$

such that the run from $((p, q, r), \sigma)$ to $((p'', r'', r''), \sigma)$ is associated with a well-nested word. By Lemma 5, we obtain $p'' = p$ and $r'' = r$. Considering the return transition associated with r, and by definition of δ'_c and δ'_r, there exists a call transition

$((p', q'', r'), c, (\gamma, p', r'), (p, p, r)) \in \delta'_c$ for some letter $c \in \Sigma_c$. In addition, as $(p, q, r) \in$ WN, there is a run $(p, \sigma) \to^* (q, \sigma) \to^* (r, \sigma)$ in A, and thus by Lemma 7 we have a run $((p, p, r), \sigma) \to^* ((p, q, r), \sigma)$. As the top symbol of σ is (γ, p', r'), the above call transition can be used. As a consequence, we have proven that there exists a run from configuration $((p', q'', r'), \sigma')$ to configuration co. In particular, configuration $((p', q'', r'), \sigma')$ is a co-reachable configuration whose height is n, and the result follows by induction hypothesis.

By construction, some useless state may remain in A': indeed, A' may contain some state q such that for no stack σ, (q, σ) is reachable (and thus co-reachable). However, deciding whether for some stack σ, (q, σ) is reachable can be reduced to the emptiness problem of pushdown automata accepting on final states. Since this problem is decidable in polynomial time, We can easily remove useless states from A'. It follows that A' is trimmed. $\square$

To summarize the results presented in this section, we have the following theorem:

Theorem 9. *Let A be a* VPA. *Then:*

- $\mathsf{trim}(A)$ *can be built in polynomial time,*
- $L(A) = L(\mathsf{trim}(A))$*, and more precisely, there exists a mapping from transitions of A' to those of A which yields a bijection between accepting runs of A' and those of A,*
- $\mathsf{trim}(A)$ *is trimmed.*

4 Application to VPA with weights

We show in this section that our constructions can be applied to VPA with weights, for instance visibly pushdown transducers (see [3]) where transitions are in addition labelled with output words, and VPA with multiplicities ($\mathbb{N}$-VPA for short), where transitions are labelled by integers.

We consider a monoid $(M, \cdot)$ which will be used to represent weights associated with transitions (for instance Σ^* equipped with concatenation in the case of transducers and $\mathbb{N}$ equipped with addition for $\mathbb{N}$-VPA).

Definition 10 (Weighted VPA). *A* weighted visibly pushdown automaton *on finite words over Σ with weights in $(M, \cdot)$ is a tuple $A = (Q, I, F, \Gamma, \delta)$ where Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ the set of final states, Γ is a finite stack alphabet, $\delta = \delta_c \uplus \delta_r \uplus \delta_\iota$ is the (finite) transition relation, with $\delta_c \subseteq Q \times \Sigma_c \times M \times \Gamma \times Q$, $\delta_r \subseteq Q \times \Sigma_r \times M \times \Gamma \times Q$, and $\delta_\iota \subseteq Q \times \Sigma_\iota \times M \times Q$. Given a transition $t \in \delta$, we denote by $\lambda(t) \in M$ its weight.*

The notions of configurations and runs are lifted from VPA to weighted VPA. Given a run ρ over a sequence of transitions $\eta = (t_i)_{1 \leq i \leq k}$, we define the weight of ρ, denoted $\langle \rho \rangle$, as $\langle \rho \rangle = \Pi_{i=1}^{i=k} \lambda(t_i)$.

Then, the behaviour of the weighted VPA A is represented by the formal power serie $\langle\langle A \rangle\rangle$ from Σ^* to multisets over M, defined by $\langle\langle A \rangle\rangle(u) = \{\{\langle \rho \rangle \mid \rho \in Runs(u)\}\}$.

In the sequel, we are interested in the following property of a weighted VPA:

$$(\dagger) \iff \begin{cases} (p,c,m,\gamma,q),(p,c,m',\gamma,q) \in \delta_c \Rightarrow m = m' \\ (p,r,m,\gamma,q),(p,r,m',\gamma,q) \in \delta_r \Rightarrow m = m' \\ (p,i,m,q),(p,i,m',q) \in \delta_\iota \Rightarrow m = m' \end{cases}$$

Intuitively, this property states that two transitions that coincide on their source and target states, input and stack symbols, have the same weight. We prove that any weighted VPA can be turned into an equivalent weighted VPA verifying this property:

Proposition 11. *Any weighted* VPA *can be turned into an equivalent weighted* VPA *verifying property* $(\dagger)$.

Proof. Let $A = (Q, I, F, \Gamma, \delta)$ be a weighted VPA. We build an equivalent weighted VPA $A' = (Q', I', F', \Gamma', \delta')$ that satisfies property $(\dagger)$. Intuitively, the construction stores in the location the last transition that has been fired. This allows to distinguish between two transitions that are equivalent w.r.t. the underlying VPA, but which have different weights. Indeed, in this new model, they are not equivalent anymore w.r.t. the underlying VPA.

Let $\#$ be a new symbol, and $\delta_\# = \delta \cup \{\#\}$. The symbol $\#$ is used as an initialization, as no transition has been fired yet. We define A' as follows:

- $Q' = Q \times \delta_\#$
- $I' = I \times \{\#\}$
- $F' = F \times \delta_\#$
- $\Gamma' = \Gamma$
- δ' is defined as follows: (similarly for δ'_r, δ'_ι)

$$((p,t),c,m,\gamma,(q,t')) \in \delta'_c \iff t' = (p,c,m,\gamma,q) \in \delta_c \wedge t \in \delta_\#$$

The correctness of this construction can be proven by an induction on the length of runs. □

Property $(\dagger)$ implies that the weight of a run of a weighted VPA only depends on the run of the underlying VPA. This result allows to apply the trimming procedure presented in this paper to any weighted VPA. Indeed, we have proven that there exists a mapping from transitions of $\mathsf{trim}(A)$ to those of A yielding a bijection between accepting runs. Thus, our construction preserves the weight associated to each word. Formally, we have:

Proposition 12. *Let A be a weighted* VPA*. We can build in polynomial time a weighted* VPA *A' that is trimmed and such that $\langle\langle A\rangle\rangle = \langle\langle A'\rangle\rangle$.*

References

1. R. Alur and P. Madhusudan. Visibly pushdown languages. In *STOC*, pages 202–211, 2004.
2. R. Alur and P. Madhusudan. Adding nesting structure to words. *JACM*, 56(3):1–43, 2009.
3. E. Filiot, J.-F. Raskin, P.-A. R. Reynier, F. Servais, and J.-M. Talbot. Properties of visibly pushdown transducers. In *Proc. MFCS'10*, volume 6281 of *LNCS*, pages 355–367. Springer, 2010.
4. D. Girault-Beauquier. Some results about finite and infinite behaviours of a pushdown automaton. In *Automata, Languages and Programming*, volume 172 of *Lecture Notes in Computer Science*, pages 187–195. Springer, 1984.